CS150 Exam #1 Review Topics

1) What are keywords? Can you give some examples?
2) What is the standard output? How do you display something on the standard output?
3) What is the standard input?
4) What is the difference between the insertion operator `<<` and the extraction operator `>>`? Specifically, when and how is each used?
5) How do you specify comments in C++?
6) What is `main`?
7) What are preprocessor directives? Can you give an example?
8) What is the difference between a declaration statement and an executable statement? Can you give an example?
9) What are operands?
10) What is a string literal? What is a string variable? How do you display each on the screen?
11) What is a char literal? What is a char variable? How do you display each on the screen?
12) When do we use CONST in a variable declaration? How do we use it?
13) What is an escape character? How do you use them? Be able to explain what each of the characters \n, \t, \\, and \" do (when used appropriately).
14) Know the different datatypes from C++ that we have discussed, including `int`, `float`, `double`, `string`, `char`, and `bool`. Can you give examples of when you might use each datatype? How do you declare a variable of a given datatype?
15) What is the difference between a variable declaration and variable assignment? Can you give examples?
16) What are the rules for variable identifiers? Can you give examples of expressions that will work as variable identifiers and ones that will not?
17) Know the five main arithmetic operators and how to use them. Also know the issues that may arise when doing division with integers.
18) What does operator precedence mean? Can you give an example? Do you know where various operations fall within the operator precedence hierarchy? Can you evaluate an expression that contains a combination of operators in the operator precedence hierarchy?
19) What does operator associativity mean? Can you give an example?
20) What is a relational operator? What are they used for? Can you give an example?
21) Know how to implement a single-alternative `if` statement.
22) Be able to write a basic but complete C++ program.